CLAIM AMENDMENTS

This listing of claims will replace all prior versions and listings of claims in the application.

1. (Currently Amended) A method of performing a reduction operation in a cryptographic calculation in a digital computer, the method comprising:

selecting a modulus having a first section with a plurality of "1" Most Significant Word states and a second section which further comprises: a plurality of "1" or "0" states whereby a number formed of the two sections is a modulus; and

operating a reduction operation on the modulus comprising:

multiplying a first variable $n_{0'}$ by a second variable r_3 to produce a first result;

adding the first result to a third variable r_1 and B multiplied by a fourth variable ~~Br_2~~ r_2 to produce a first sum, wherein the first sum corresponds to a first equation: $n_{0'}\, r_3 + B\, r_2 + r_1$;

dividing the first sum into an upper half and a lower half;

multiplying the upper half by the first variable $n_{0'}$ to produce a second result;

adding the second result to the lower half and a fifth variable r_0 to produce a second sum, thereby permitting use of the second sum as the modulus; and

using the modulus in the cryptographic calculation.

2. (Currently Amended) ~~A~~ The method ~~according to~~ of claim 1, further comprising:

effecting a plurality of multiplication operations.

3. (Currently Amended) ~~A~~ The method ~~according to~~ of claim 2, further comprising:

effecting a plurality of multiplication operations followed by effecting a reduction operation.

4. (Currently Amended) ~~A~~ The method ~~according to~~ of claim 3, further comprising:

repeating the ~~combined multiplication operations and reduction operation~~ effecting step of claim 3.

5. (Currently Amended) ~~A~~ The method ~~according to~~ of claim 1, further comprising:

using a multiple of the modulus.

6. (Currently Amended) ~~A~~ The method ~~according to~~ of claim 1, further comprising:

~~wherein, when a last multiplication gives an overflow, the overflow is added~~ adding an overflow from a last multiplication to a part of a selected number.

7. (Currently Amended) ~~A~~ The method ~~according to~~ of claim 6, further comprising:

~~wherein, when the overflow addition step produces an overflow, then~~ adding the first variable n_0' ~~is added~~ to the overflow.

8. (Currently Amended) ~~A~~ The method ~~according to~~ of claim 1, wherein a carry c between two adjacent multiplications is effected as an addend in ~~the next~~ a subsequent multiplication.

9. (Currently Amended) ~~A~~ The method ~~according to~~ of claim 1, further comprising:

monitoring ~~the~~ a number of leading "1"s to determine if the number is less than (k-2).

10. (Currently Amended) ~~A~~ The method ~~according to~~ of claim 9, further comprising:

initiating a next calculation~~when the number of leading "1"s is less than (k-2)~~.

11. (Currently Amended) ~~A~~The method ~~according to~~ of claim 1, the method further comprising:

operating 192-bit ECC and a word size of 64-bit,

the modulus comprises a first section of 138 bits and a second section of 54 bits.

12. (Currently Amended) ~~A~~The method ~~according to~~ of claim 1, the method further comprising:

operating 128-bit ECC and a word size of 64-bit,

the modulus comprises a first section of 74 bits and a second section of 54 bits.

13. (Currently Amended) ~~A~~The method ~~according to~~ of claim 1, the method further comprising:

operating 256-bit ECC and a word size of 64-bit,

the modulus comprises a first section of 202 bits and a second section of 54 bits.

14. (Currently Amended) A computer program product directly loadable into the internal memory of a digital computer, comprising:

software code portions for performing the method of claim 1 ~~when said product is run~~ on a computer.

15. (Currently Amended) A computer program directly loadable into the internal memory of a digital computer, comprising:

software code portions for performing the method of claim 1 ~~when said program is run~~ on a computer.

16-17. (Canceled).

18. (Currently Amended) An apparatus that performs a reduction operation in a cryptographic calculation on a digital computer, the apparatus comprising:

a plurality of input registers that store a plurality of input operands;

a plurality of output registers that store a plurality of outputs; and

a multiplier that produces said outputs using a function that operates on variables from both said input registers and said output registers; wherein said multiplier selects a modulus having a first section with a plurality of "1" states and a second section having a plurality of "1" or "0" states whereby a number formed of the two sections is a modulus and performs a reduction operation on the modulus, the reduction operation comprising:

multiplying a first variable $n_{0'}$ by a second variable r_3 to produce a first result;

adding the first result to a third variable r_1 and B multiplied by a fourth variable ~~Br_2~~ r_2 to produce a first sum, wherein the first sum corresponds to a first equation: $n_{0'} r_3 + B\, r_2 + r_1$;

dividing the first sum into an upper half and a lower half;

multiplying the upper half by the first variable $n_{0'}$ to produce a second result; adding the second result to the lower half and a fifth variable r_0 to produce a second sum, thereby permitting use of the second sum as the modulus; and

using the modulus in the cryptographic calculation.

19. (Previously Presented) The apparatus of claim 18, further comprising:

means to effect a plurality of multiplication operations.

20. (Previously Presented) The apparatus of claim 19, further comprising:

means to effect a plurality of multiplication operations followed by a reduction operation.

21. (Previously Presented) The apparatus of claim 20, further comprising:

means to repeat the plurality of multiplication operations and the reduction operation.

22. (Previously Presented) The apparatus of claim 18, further comprising:

means to use a multiple of the modulus.

23. (Currently Amended) The apparatus of claim 18, further comprising:

means~~, when a last multiplication gives an overflow, to add the overflow to a part of a selected number~~ to add an overflow from a last multiplication to part of a selected number.

24. (Currently Amended) The apparatus of claim 23, further comprising:

means~~, when the overflow addition step produces an overflow, to add~~ to add the first variable $n_{0'}$ to the overflow.

25. (Currently Amended) The apparatus of claim 18, further comprising:

means to effect a carry c between two adjacent multiplications as an addend in ~~the next~~ a subsequent multiplication.

26. (Currently Amended) ~~Apparatus according to~~ The apparatus of claim 18, further comprising:

means to monitor ~~the~~ a number of leading "1"s to determine if the number is less than (k-2).

27. (Currently Amended) The apparatus of claim 26, further comprising:

means to initiate a next calculation ~~when the number of leading "1"s is less than (k-2)~~.

28. (Currently Amended) The apparatus of claim 18, ~~further comprising:~~

~~with means for 192-bit ECC and a word size of 64-bit,~~

wherein the modulus comprises a first section of 74 bits and a second section of 54 bits.

29. (Currently Amended) The apparatus of claim 18, ~~further comprising:~~

~~with means for 128-bit ECC and a word size of 64-bit,~~

wherein the modulus comprises a first section of 74 bits and a second section of 54 bits.

30. (Currently Amended) The apparatus of claim 18, ~~further comprising:with~~

~~means, for 256-bit ECC and word size of 64-bit,~~ wherein the modulus comprises a first section of 202 bits and a second section of 54 bits.

31-33. (Canceled).